Bullseye: A Stock Market Visualization and Prediction Service

Vipul Gharde
Graduate Student in Computer Science at Rutgers University

Table of Contents

Abstract

This paper presents Bullseye, a web-based stock market visualization and prediction service that leverages machine learning and deep learning algorithms to provide users with real-time market data, customizable visualizations, and predictive analytics. The service is designed to help investors make informed investment decisions by identifying investment opportunities based on forecasted market trends. To evaluate the effectiveness of Bullseye, various LSTM models were trained and tested on 5-year historical closing stock market data from Apple. The best-performing model achieved a validation RMSE of 5.34 and a validation R^2 value of 0.71. The results demonstrate the potential of Bullseye to improve prediction accuracy and provide valuable insights into market trends. Overall, Bullseye offers a user-friendly web interface and a continuously updated algorithm to help investors gain valuable insights into the stock market and make informed investment decisions.

1. Introduction

Bullseye is a web service designed to provide a more efficient and effective way of visualizing and predicting stock market trends. The stock market is a complex and volatile environment, which can make forecasting trends a challenging task. However, Bullseye utilizes machine learning and deep learning algorithms to provide users with real-time market data, predictive analytics, and customizable visualizations through a user-friendly web interface. This paper aims to evaluate the effectiveness of Bullseye in predicting stock trends by analyzing historical stock data.

In today's fast-paced world, keeping track of the stock market and making informed investment decisions can be a daunting task, particularly for those without extensive knowledge of the stock market. Bullseye seeks to address this challenge by providing users with an easy-to-use interface that provides valuable insights into market trends and identifies investment opportunities.

Compared to existing services, Bullseye stands out due to its ability to leverage the power of machine learning and deep learning algorithms to make accurate predictions about stock market trends. With its customizable visualizations and real-time market data, users can quickly gain insights into market trends, helping them make more informed investment decisions.

The rest of this paper discusses related work in Section 2, followed by a description of Bullseye's implementation in Sections 3-6. The effectiveness of Bullseye in predicting stock trends are then evaluated using historical stock data in Section 7. Finally, the claims are described in Section 8.

2. Cross-reference to related work

Several existing services aim to provide users with insights into the stock market, such as Yahoo Finance, Google Finance, and Bloomberg. These services provide historical and real-time stock data, news, and customizable visualizations to users. However, these services often require users to have a significant amount of knowledge about the stock market to interpret the data accurately. Additionally, these services typically use traditional statistical methods to analyze stock data and do not incorporate advanced machine learning and deep learning algorithms, which can limit their accuracy in predicting market trends.

Other services, such as Alpha Vantage and Quandl, offer machine learning-based predictions of stock trends. However, these services may require users to have a strong background in machine learning to utilize them effectively. Moreover, these services may only provide predictions for specific stocks or markets, limiting their applicability to a broader range of users.

Compared to these existing services, Bullseye offers several advantages. Bullseye incorporates advanced machine learning and deep learning algorithms to analyze stock data and provide accurate predictions of market trends. Additionally, Bullseye's user-friendly interface makes it accessible to users without extensive knowledge of the stock market or machine learning. Bullseye also provides real-time market data, predictive analytics, and customizable visualizations to users, making it a comprehensive solution for investors and analysts looking to make informed decisions about the stock market.

3. Background of the service

Field of the Service:

Bullseye is a web service that falls under the field of financial technology, or fintech. Fintech refers to the use of technology to provide financial services, including banking, investing, and

trading. Fintech has been gaining popularity in recent years due to advancements in technology and increasing demand for more accessible and user-friendly financial services.

Description of the Related Art:

Prior to the development of Bullseye, several existing services aimed to provide users with insights into the stock market. However, these services often required extensive knowledge of the stock market and traditional statistical methods to interpret the data accurately. Furthermore, these services did not incorporate advanced machine learning and deep learning algorithms, which can limit their accuracy in predicting market trends.

Several fintech companies have recently emerged, incorporating machine learning and other advanced technologies to improve financial services. For example, robo-advisors use algorithms to provide automated investment advice, and blockchain technology has been used to improve security and transparency in financial transactions. However, these services may still require a significant amount of expertise to utilize effectively.

Bullseye aims to bridge the gap between traditional financial services and advanced fintech solutions by providing users with real-time market data, predictive analytics, and customizable visualizations through a user-friendly interface. By utilizing machine learning and deep learning algorithms, Bullseye can provide users with accurate predictions of market trends, making it a valuable tool for investors and analysts looking to make informed decisions about the stock market.

4. Brief summary of the service

Bullseye is a web service that provides users with an efficient and effective way to visualize and predict stock market trends. It utilizes advanced machine learning and deep learning algorithms, coupled with real-time market data and customizable visualizations, to give users accurate predictions of market trends. Bullseye's user-friendly interface makes it accessible to a wide range of users, even those without extensive knowledge of the stock market. With Bullseye, users can gain valuable insights into market trends and make informed investment decisions.

5. Brief description of the several views of the drawing

The Bullseye web service provides a user-friendly interface for viewing stock data and predictions. The navigation bar and footer of the website are made using Bootstrap, while the logo is an SVG file created with Figma. The homepage features a search bar in the center where users can search for a stock ticker to view its historical stock data and predictions, as shown in Figure 1.

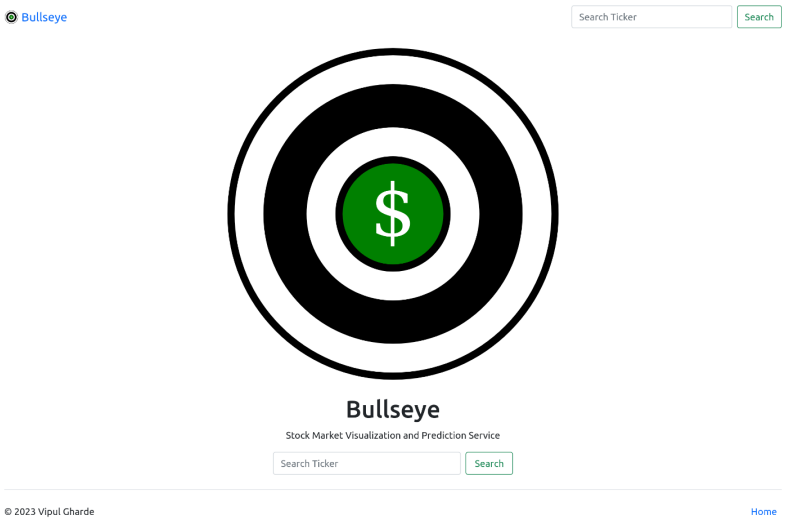

Figure 1: Bullseye homepage

One of the key features of the Bullseye web service is the interactive plot that allows users to visualize the stock price history and volume of stocks traded for a selected ticker, as shown in Figure 2. The plot is built using Plotly. The title of the visualization page includes the company name, current market price, relative price from the previous day, and percentage increase or decrease in stock price from the previous day. The relative price and percentage increase/decrease are color-coded to indicate whether the stock price increased (green) or decreased (red) from the previous day. The page also displays the prediction of the stock's closing price for the next working day, which is color-coded to indicate whether the closing stock price will increase (green) or decrease (red) in the next working day.

Users can view the stock history for various time periods, ranging from 1 day to 5 years in the past, by clicking a button. The plot displays a candlestick chart by default, but users can choose to view the Open, High, Low, or Close line plots instead. Additionally, users can view the plots of moving averages of varying window sizes by clicking the appropriate legends below the plot. The volume of stocks traded is displayed alongside the stock price history.

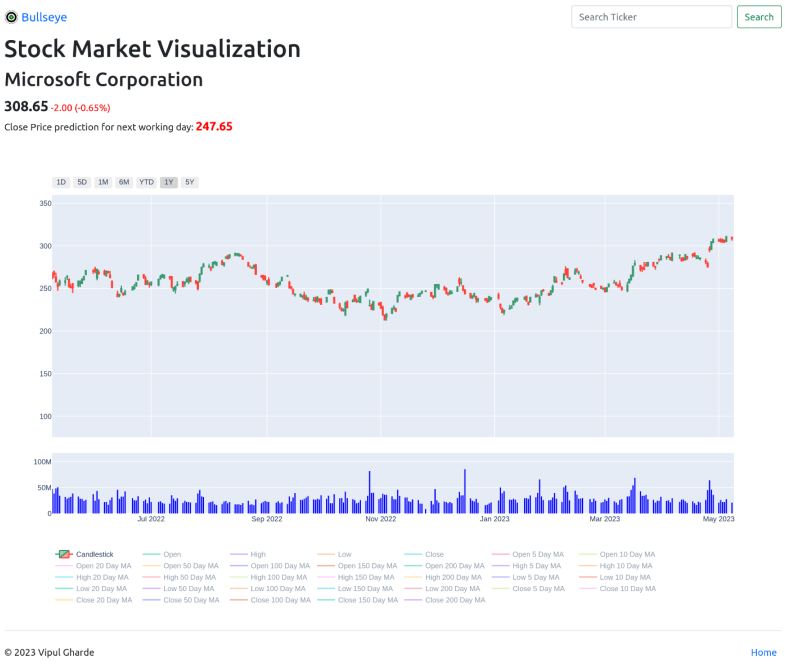

Figure 2: Visualization of Microsoft Corporation stocks from the past year

6. Detailed description of the web service

The web service is developed using the Django Python library, which provides a framework for building web applications quickly and efficiently. The logo is an SVG file created with Figma, and the stock-related data for a particular ticker is fetched using the yfinance Python library. The stock data is then visualized using Plotly, which provides an easy-to-use interface for creating interactive and dynamic data visualizations.

To use the web service, the user can access the web application through a web browser. From there, they can search for a ticker in the search bar to view the stock price history and the volume of stocks traded for the same day, as depicted in a candlestick and bar plot respectively. The visualization page's title displays the company name, current market price, relative price from the previous day, and percentage increase or decrease in stock price from the previous day. The relative price and percentage increase/decrease are color-coded to show whether the stock price has increased (green) or decreased (red) from the previous day. Additionally, the page displays the stock's predicted closing price for the next working day, with colors indicating whether the closing stock price will increase (green) or decrease (red) in the next working day.

Users can also choose to view the stock history for different periods, such as 1 day, 5 days, 1 month, 6 months, year-to-date, 1 year, or 5 years in the past, by selecting the appropriate button. Furthermore, they can view line plots of the Open, High, Low, or Close data instead of the candlestick plot. Users can also view the plots of moving averages of varying window sizes by clicking the appropriate legends beneath the plot.

The web service's closing price predictions are generated by an LSTM model trained on Apple's 5-year closing price data from 2018-01-01 to 2023-01-01, using Keras. The LSTM model takes the last 60 days' closing stock data, scales it using the StandardScaler scaler from the scikit-learn library, and then uses it as input to predict the next working day's close price. The trained LSTM model and the StandardScaler scaler are saved as Pickle files within the Django app, which are loaded into memory for prediction each time there is a request to view the visualization page for a particular ticker. For prediction, the last 60 days' closing stock price data is fetched using the yfinance library, scaled, and fed into the loaded LSTM model. The visualization page is then rendered with the prediction.

The detailed description provided should be sufficient for any person skilled in the pertinent art or science to develop a similar web service without involving extensive experimentation. The best mode of using the web service is to search a ticker from the search bar, select the desired date range and plot type, and explore the stock data by hovering over the plot.

During the development of the web service, some challenges were encountered, such as handling errors that could occur when fetching data from yfinance, optimizing the speed of loading the data and plotting, and ensuring that the web service is responsive and easy to use. However, these challenges were overcome by testing and debugging the code, and the resulting web service is a user-friendly and reliable tool for visualizing stock data.

7. Evaluation

The evaluation section of this paper aims to demonstrate the effectiveness of the Bullseye web service in predicting stock market trends. To evaluate the performance of the service, 5-year historical closing stock market data from Apple from 2018-01-01 to 2023-01-01 was collected and used to train and test the LSTM models.

To design the experiments, the data was split into training and validation sets in an 80:20 ratio to evaluate the models' performance on the dataset. Various LSTM models were trained by adjusting the number of layers and hyperparameters. The models were then evaluated on the validation set, and their performances were compared based on their training loss, validation RMSE, and validation R^2 values.

The LSTM model with the best performance consisted of 2 LSTM layers, with 64 and 32 units, respectively, followed by 2 dense layers with 25 and 1 unit respectively. The model uses 60 days of historical closing prices, which are scaled using the StandardScaler scaler from the scikit-learn library before being fed as input. The optimizer chosen was adam, with the mean_squared_error loss function. The model achieved a training loss of 0.0153, a validation RMSE of 5.34, and a validation R^2 value of 0.71.

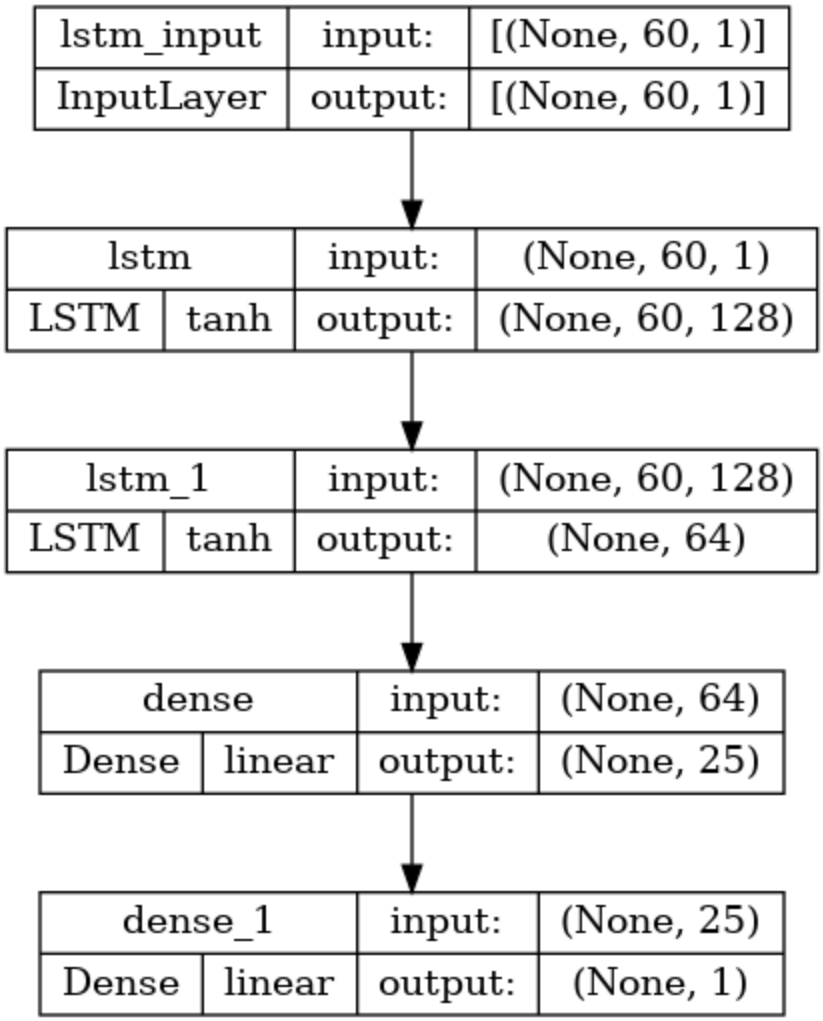

Figure 3: Architecture of the Best Performing LSTM Model

However, there are some limitations to the evaluation presented in this paper. Firstly, training the model on historical stock data from Apple may not be the best approach for predicting the next working day's closing price for another stock. Instead, each company or companies that fall within each industry should have its own model for predicting the next working day's closing price. This is because each company's stock prices are affected by unique factors, such as industry trends, company performance, news, and global economic conditions, which may not be captured by a model trained on data from a different company.

Second, using a regression model to predict exact stock prices may not always be the best approach as it can be highly volatile and unpredictable. Instead, using a classification model to predict whether the closing price for the next day is higher or lower than today's may be a better approach.

Additionally, some feature engineering could be done to enhance the model's predictive power. For instance, creating features such as the difference between the opening and closing price for the day, the difference between the low and high price for the day, and a binary indicator indicating whether it is the end of the quarter or not could improve the model's performance. This binary indicator could be useful because the quarterly results of a company can heavily affect its stock prices, as every company prepares its quarterly results and publishes them publicly so that people can analyze the company’s performance.

8. Claims

1. A web-based stock market trend visualization and prediction service, comprising: a) a machine learning and deep learning algorithm trained on historical stock data for forecasting stock trends, b) a user-friendly web interface that provides real-time market data and customizable visualizations, and c) a predictive analytics system that enables users to gain valuable insights into market trends and make informed investment decisions.
2. The web service of claim 1, wherein the machine learning and deep learning algorithm is trained on historical stock data to improve prediction accuracy.
3. The web service of claim 1, wherein the predictive analytics system includes tools for identifying investment opportunities based on forecasted market trends.
4. The web service of claim 1, wherein the user-friendly web interface includes interactive visualizations for exploring stock market data and trends.
5. The web service of claim 1, wherein the machine learning and deep learning algorithm and predictive analytics system are continuously updated to ensure the accuracy of predictions and insights.

References

[1] Ran Aroussi. (2021). yfinance: Yahoo! Finance market data downloader. Available at: https://pypi.org/project/yfinance/

[2] Django. (n.d.). The web framework for perfectionists with deadlines. Retrieved from https://www.djangoproject.com/

[3] Plotly. (n.d.). Collaborative data science. Retrieved from https://plotly.com/python/

[4] Figma. (n.d.). Design, prototype, and collaborate all in one place. Retrieved from https://www.figma.com/

[5] Bootstrap. (n.d.). Build responsive, mobile-first projects on the web with the world's most popular front-end component library. Retrieved from https://getbootstrap.com/

[6] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830. Available at: https://scikit-learn.org/stable/

[7] Chollet, F. (2015). Keras. Retrieved from https://keras.io/

[8] Lipton, Z. C., Kale, D. C., & Elkan, C. (2015). Learning to diagnose with LSTM recurrent neural networks. arXiv preprint arXiv:1511.03677.

[9] Zhang, X., & Lai, K. K. (2018). Stock price prediction using LSTM, RNN and CNN-sliding window model. Journal of Physics: Conference Series, 1036(1), 012014.